

本書補足情報

本書（第3刷・第2刷・第1刷）の掲載内容に、以下の補足情報がございます。

(2020年3月11日更新)

P.264 下から1番目のコードの直後

本来であれば、マクロ内で使用しているデータの定義「db i」を「%else」に含める必要はありません。しかしながら、NASMの一部のバージョンでは「%exitrep」を実行してもすぐに繰り返し処理を終了しないので、すべてのバージョンで同じ出力結果を得るために、このように記載しています。

P.593 上から1番目のコードの直後

同様の修正は、線の描画関数（draw_line）にも行います。線の描画関数は点の描画関数を繰り返し呼び出しているため、ループカウンタとして使用している ECX レジスタの値を変更しないように注意する必要があります。

```
prog/src/modules/protect/draw_line.s
draw_line:
    ...
#ifdef USE_SYSTEM_CALL
    mov     eax, ecx                ; // 繰り返し回数を保存

    mov     ebx, [ebp +24]          ; EBX = 表示色;
    mov     ecx, [ebp - 8]          ; ECX = X座標;
    mov     edx, [ebp -20]          ; EDX = Y座標;
    int     0x82                    ; sys_call(1, X, Y, 色, 文字);

    mov     ecx, eax
#else
    cdecl   draw_pixel, dword [ebp - 8], ¥
           dword [ebp -20], ¥
           dword [ebp +24]        ; // 点の描画
#endif
```

P.659 「fat.s」のコード以降

```
prog/src/43_fat/fat.s
;*****
;   FAT:データ領域
;*****
        times FILE_START - ($ - $$) db 0x00
;-----
FILE:   db    'hello, FAT!'
.end:   db    0

ALIGN 512, db 0x00

        times (512 * 63)    db 0x00
```

作成したファイルはカーネルの末尾にインクルードして使います。

```
prog/src/43_fat/kernel.s
;*****
;   パディング
;*****
        times KERNEL_SIZE - ($ - $$) db 0x00    ; パディング
;*****
;   FAT
;*****
#include "fat.s"
```

FAT はカーネルの直後に配置されるので、1 番目の FAT 領域の開始位置はカーネルサイズと同じ値となります。2 番目の FAT 領域は BPB の「FAT 領域のセクタ数(0x16)」に 256、「セクタのバイト数(0x0B)」に 512 を設定したので、それよりも 0x02_0000 (256 セクタ×512) バイト後方に配置します。

```
prog/src/include/define.s

FAT_SIZE      equ    (1024 * 128)    ; FAT-1/2
ROOT_SIZE     equ    (1024 * 16)     ; ルートディレクトリ領域

FAT1_START    equ    (KERNEL_SIZE)
FAT2_START    equ    (FAT1_START + FAT_SIZE)
ROOT_START    equ    (FAT2_START + FAT_SIZE)
FILE_START    equ    (ROOT_START + ROOT_SIZE)
```

これまでに作成したディスクイメージは、次の図のように配置されています。